# PARAMETRIC SHAPE GRAMMAR INTERPRETER

Inventors: Jay McCormack and Jonathan Cagan

## CROSS-REFERENCE TO RELATED APPLICATIONS

Not Applicable.

## STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Certain of the research leading to the present invention was sponsored by the United States National Science Foundation under contract No. DMI-9713782. The United States Government may have rights in the invention.

## BACKGROUND OF INVENTION

### Field of Invention

The present invention relates generally to shape grammars and, more particularly, to shape grammar systems and methods having parametric shape recognition.

### Description of the Background

A shape grammar is a set of rules, based on shape, that is used to generate designs through rule applications. Rules take the form of $a \rightarrow b$, where $a$ and $b$ both denote shapes. A rule is applicable if the left-hand shape, $a$, can be found in the design shape, denoted $c$. If the

rule is applied, the left hand shape is subtracted from the design and the right-hand shape is added to the design, denoted $c - \tau(a) + \tau(b)$, where shapes $a$ and $b$ undergo a transformation $\tau$ according to the transformation required to make shape $a$ a subshape of shape $c$.

Shape grammars, having their roots in architecture literature, have recently found application in engineering, such as in the design of coffeemakers, lathe process plans, roof trusses, and microelectromechanical systems (MEMS) resonators. Shape grammars may be conceptualized of as a type of expert system based on geometry. Shape grammars, however, have succeeded in engineering applications where traditional expert systems have failed because of: (i) their direct handing of reasoning about geometry; (ii) their ability to operate on a parametric geometric representation; and (iii) their ability to support emergence of shape. These advantages presage a future in which shape grammars play an increasingly larger role in engineering design in comparison with the traditional expert systems.

In the past, however, shape grammars have been limited by the difficulty and time intensity in their implementations. Implementations have not allowed for general parametric shape recognition. Engineering shape grammars in particular have been restricted to limited, non-parametric shape recognition and often are hard-coded. These drawbacks minimize much of the beneficial potential of shape grammars.

Accordingly, there exists a need for a shape grammar system that uses shape recognition to provide, for example, an automated approach to product generation. There further exists a need for a shape grammar system in which engineering knowledge (geometry-based and otherwise) may be incorporated into implementation design rules in order to drive design exploration and the generation of designs toward a desired end.



3

## BRIEF SUMMARY OF INVENTION

The present invention is directed to a method of recognizing a first shape in a second shape. According to one embodiment, the method includes decomposing the first shape into at least one subshape belonging to one of a plurality of subshape groups, and searching the second shape for a parametric transformation of the subshape.

According to another embodiment, the present invention is directed to a shape grammar interpreter, including a shape decomposition module, and a shape recognition module in communication with the shape decomposition module.

The present invention allows for shape grammars, including engineering shape grammars, to be implemented in a fraction of the time that it currently takes to hard code them. Consequently, the present invention allows shape grammars to be adjusted, fine tuned, and adapted to the changing design scenario presented to the rule writer. The shape grammar interpreter of the present invention therefore possesses the features desired in an engineering grammar implementation, including general parametric shape recognition, providing designers with the possibility of exploring the promising potential of engineering shape grammar systems. These and other benefits of the present invention will be apparent from the detailed description hereinbelow.

## DESCRIPTION OF THE FIGURES

For the present invention to be clearly understood and readily practiced, the present invention will be described in conjunction with the following figures, wherein:



4

Figure 1 is a block diagram of a shape grammar system according to one embodiment of the present invention;

Figures 2 and 3 are diagrams of examples of line segments belonging to subshape groups according a default hierarchy of subshape groups according to one embodiment of the present invention;

Figure 4 is a block diagram of the process flow through the parametric shape grammar interpreter of the shape grammar system of Figure 1 according to one embodiment of the present invention;

Figures 5-11 are diagrams illustrating a method of shape decomposition according to one embodiment of the present invention;

Figures 12-19 are diagrams illustrating a method of parametric shape recognition according to one embodiment of the present invention;

Figures 20-23 are diagrams illustrating a method of using parametric shape recognition to apply a given shape grammar rule to a given initial design shape according to one embodiment of the present invention; and

Figures 24-27 are diagrams illustrating a method of using parametric shape recognition to apply a set of shape grammar rules to a given initial design shape according to another embodiment of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is a block diagram of a shape grammar system 10 according to one embodiment of the present invention. The shape grammar system 10 includes a parametric shape grammar interpreter 12, including a shape decomposition module 14 and a shape recognition module 16.

The shape grammar system 10 also includes a rule application module 18 and an intelligent rule selection module 20, which are in communication with the parametric shape grammar interpreter 12. The shape grammar system 10 may also include an input/output (I/O) interface module 22, as illustrated in Figure 1. The shape grammar system 10, as described hereinbelow, may be used to implement, for example, architectural shape grammars, engineering shape grammars, and industrial design shape grammars, with parametric shape recognition. The parametric shape grammar interpreter 12 will be described herein as being used to recognize the left-hand shape of a shape grammar rule in the initial design shape(s) through the steps of decomposing the shape into subshapes and progressively searching for parametric transformations of those subshapes, however, it should be recognized that the benefits of the present invention may be realized in any application requiring parametric shape recognition, and is not limited to shape grammar applications.

The system 10 may be implemented using, for example, a computer, such as a workstation or a personal computer, a microprocessor, or an application specific integrated circuit (ASIC). The modules 14, 16, 18, 20, and 22 may be implemented as software code to be executed by the system 10 using any type of computer instruction type suitable such as, for example, microcode, and can be stored in, for example, an electrically erasable programmable read only memory (EEPROM), or can be configured into the logic of the system 10. According to another embodiment, the modules 14, 16, 18, 20, and 22 may be implemented as software code to be executed by the system 10 using any suitable computer language such as, for example, C or C++ using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer readable medium, such as a



6

random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a CD-ROM.

The parametric shape grammar interpreter 12 may perform the operations necessary to determine whether any of a predefined set of shape grammar rules may be applied to a particular shape (or set of shapes). In addition, the interpreter 12 may determine how a particular rule may be applied to the shape(s). As described hereinbelow, the interpreter 12 may perform these operations by decomposing, for example, the left-hand shape of a shape grammar rule into a group of subshapes, thereby allowing for any part of the shape to be transformed with any possible transformation, although, as discussed hereinbefore, it is not limited to such shapes. The interpreter 12 may perform these operations with respect to, for example, a left-hand shape of a rule having one-dimensional, two-dimensional or three-dimensional shapes. In addition, the left-hand shape may include, for example, straight line segments, curved line segments, planes, or three-dimensional objects. Once the interpreter 12 determines whether a rule may be applied and how to apply the rule, whether the rule should be applied to the shape may be determined, for example, by a user of the system 10 or the intelligent rule selection module 20. The rule application module 18 may then apply the rule to the shape if so determined.

The shape decomposition module 14 decomposes a shape such as, for example, the left-hand shape of a rule (the shape $a$ in the rule $a \rightarrow b$) into a group of subshapes contained in the shape. The groups may be defined such that subshapes belonging to different groups do not share, for example, line segments for two-dimensional shapes. The group of shapes may be ordered according to a hierarchy of , for example, decreasing restrictions or constraints for more



7

efficient searching, as described hereinbelow, although it is not necessary for the subshape groups to be so ordered.

For an embodiment in which the subshape groups are ordered according to a hierarchy of decreasing constraints, the basis of the hierarchy of constraints may be, for example, defined by the designer or it may be a default hierarchy. A default hierarchy may be designed, for example, to interpret the designer's intentions and preferences through particular features present in a shape which defines part of a shape grammar rule. For example, the default hierarchy may be intended to separate the parts of the left-hand shape of the rule that the designer specified exactly from the parts of the shape that were intended as a general scheme.

For example, in defining a default hierarchy for an embodiment in which the left-hand shapes of the predefined shape grammar rules include shapes having straight lines in a single plane, it is recognized that there is a limited set of transformations that can be applied to the shapes, such as translation, rotation, scaling (isotropic and anisotropic), and shearing. Of the possible transformations, some will destroy certain features of the shape and some will not. For example, no amount of translation or rotation will destroy a specific feature such as, for example, a right angle, a square, or an equilateral triangle. Shearing, however, will eliminate perpendicular intersections and symmetry in a two-dimensional shape. In addition, anisotropic scaling will also destroy symmetry unless the scaling is along or perpendicular to the line of symmetry. Isotropic scaling, on the other hand, does not affect the symmetry of a shape.

In view of the properties of these transformations, an example of a default hierarchy of subshapes may be defined as follows:


8

TABLE 1

| Subshape Group | Features | Transformations |
| --- | --- | --- |
| $s_1$ | 1) lines that intersect perpendicularly and are the same length<br>2) lines that are symmetric to more than one lines that are not parallel | translation, rotation, isotropic scaling |
| $s_2$ | 1) lines that intersect perpendicularly<br>2) lines that are symmetric to one line or<br>3) more than one lines that are parallel | translation, rotation, anisotropic scaling |
| $s_3$ | intersecting lines | translation, rotation, anisotropic scaling, shearing |
| $s_4$ | none | all |

According to such a default hierarchy, subshape group $s_1$ consists of the most constrained lines. Group $s_1$ contains the line segments that intersect perpendicularly and are the same length. Additionally, the $s_1$ group also contains any line segment that is symmetric to two or more other line segments which are not parallel. Two examples of lines that meet the symmetry criteria of group $s_1$ are the sides of a square and the legs of an equilateral triangle.

Group $s_2$ consists of the next most constrained lines, containing line segments that intersect perpendicularly. Any line segment that is symmetric to another line segment is also included in group $s_2$. Accordingly, group $s_1$ is a subset of group $s_2$. Some examples of $s_2$ lines that are not also in group $s_1$ include the sides of a rectangle and the two equal legs of an isosceles triangle.

Group $s_3$ contains the line segments that intersect. Thus, subshape groups $s_1$ and $s_2$ are subsets of $s_3$. An example of three lines that are in group $s_3$ and not $s_1$ or $s_2$ are the three line segments that make up the triangle illustrated in Figure 2.

The line segments in group $s_4$ have no discernible spatial relationship to any other line segments. Thus, the line segments in group $s_4$ are essentially those not found in $s_1$, $s_2$, and $s_3$. An example of line segments that may be found in group $s_4$ are illustrated in Figure 3.

The above-described default hierarchy is but one example of a hierarchy of subshapes ordered by decreasing constraints. According to other embodiments of the present invention, the shape decomposition module 14 may search the left-hand shape of a rule according to such other subshape hierarchies. Such other hierarchies, as described hereinbefore, may be defined by a user of the system 10, or may be a default hierarchy making different assumptions about the intent of the designer through particular features present in a shape which defines part of a shape grammar rule. For example, according to one embodiment, the hierarchy may be based on an assumption that the intersection of line segments at, for example, a right angle, is intended to represent a specific design choice, and the intersection of line segments at an angle other than a right angle is intended to represent a general scheme. According to other embodiments, the hierarchy may be based on an assumption that the intersection of line segments at, for example, sixty degrees, is intended to represent a specific design choice, and the intersection of line segments at an angle other than sixty degrees is intended to represent a general scheme.

The shape recognition module 16 searches a shape, or a set of shapes, for the subshapes belonging to the subshape groups according to the transformations appropriate for that group. According to one embodiment, parametric shape recognition may be accomplished by the shape recognition module 16 by repeating a three-step process for each of the subshape groups of the decomposed left-hand shape of a rule. The three steps of the process may include: 1) finding subshapes in the design shape, 2) subtracting the subshapes from the design shape, and 3) identifying the connectivity between the subshape and the design shape and between the



10

subshapes of successive subshape groups by, for example, marking points of intersection with labels or weights to a) the overlapping points of the decomposed left-hand shapes and also to b) points in the design equal in location to the transformed, identified points in the decomposed left-hand side shape. The process is begun with a first of the subshape groups, and progressively repeated for the others. According to one embodiment, the subshape groups are of a hierarchical order of decreasing constraints, and the process is started with the most constrained group and progressively repeated with the next most constrained subshape group. Such an embodiment generally yields more efficient searching.

For example, according to such an embodiment, the initial design shape is first searched for subshapes belonging to the most constrained group. The subshape matches, found by applying the transformations appropriate for that group, are defined as a set S. The subshapes in the set S are each subtracted from the initial design shape, producing another set of shapes, denoted as the set C. According to one embodiment, the subshapes of a decomposed shape will overlap each other, if at all, only at points because the definition of the hierarchical groups may require that the subshapes share no line segments. Thus, in order to maintain the connectivity, and hence orientation, of the subshapes, the connectivity between the shapes of sets S and C is identified and maintained. The connectivity may be maintained, for example, by identifying with labels or weights the overlapping points of the decomposed left-hand shapes and the points in the initial design corresponding to the location of the transformed, identified points in the decomposed left-hand shape.

The shape recognition module 16 may repeat this process for all of the subshape groups. The shape recognition process may end when all of the decomposed parts of the left-hand shape have been found or when one of the shape searches finds no subshapes. The shape recognition

module 16 may then add each of the shapes, maintaining the connectivity between the shapes, for each of the subshape groups found in the original shape to recognize the occurrences of the left-hand shape of the rule in the original design shape. Once the shape recognition process is completed, as described hereinbelow, the rule may then be applied.

Figure 4 is a block diagram of the process flow through the parametric shape grammar interpreter 12 according to one embodiment of the present invention. The process begins at block 30 with a determination of whether a rule remains in a set of shape grammar rules for which the left-hand shape of the rule has not been searched in the set of shapes $C_0$. The set of shape grammar rules may be defined and input to the system 10 by a user of the system 10 and may be, for example, architectural shape grammar rules, engineering shape grammar rules, or industrial design shape grammar rules. The set of rules may include one or a multitude of rules. In addition, the set of shapes $C_0$ may include one shape or a multitude of shapes. If the set does not contain any such rules, the process flow continues to block 32, and the operation of the shape grammar interpreter 12 is terminated.

Conversely, if the set does contain such a rule, the process flow continues to block 34, where the rule is selected to be applied, if applicable as determined by the parametric shape grammar interpreter 12, to the set of shapes $C_0$. From block 34, the process flow advances to block 36, where a counter, denoted as i, is set to a value of one. In addition, at block 36, the set of shapes $S_0$, as discussed hereinbelow, is set to null.

From block 36, the process advances to block 38, where the left-hand shape of the rule is decomposed into a number, denoted N, of subshape groups, denoted $s_{i...N}$. The subshape groups may be defined such that no subshapes of the decomposed left-hand shape share, for example, the same line segment. According to one embodiment, the subshape groups $s_{i...N}$ may be of a

hierarchical order of decreasing constraints, such as the default hierarchy described hereinbefore with respect to Table 1, or the hierarchy may be defined by a user of the system 10. According to other embodiments, the subshape groups are not ordered according to a hierarchical order.

From block 38, the process continues to block 40, where it is determined whether the subshape group $s_i$ is null. This corresponds to a determination of whether the left-hand shape of the rule includes a subshape belonging to the $s_i$ subshape group. For example, where i=1, it is determined whether the left-hand rule includes a subshape of the $s_1$ group. If the group $s_i$ is null, the process advances to block 42, where the set of shapes $S_i$, as described further hereinbelow, is set to null. In addition, at block 42, the set of shapes $C_i$, as described hereinbelow, is set to the same as the set $C_{i-1}$.

From block 42, the process flow advances to block 43, where it is determined whether i=N. If i does not equal N, then the process flow continues to block 44, where the counter (i) is incremented by one, and the process flow returns to block 40 such that it may be determined whether the subshape group $s_{i+1}$ is null. Conversely, if it is determined that i equals N, then the process flow advances to block 59.

If at block 40 it is determined that the $s_i$ subshape group is not null, the process flow continues to block 46, where the set of shapes $C_{i-1}$ is searched for subshapes belonging to the subshape group $s_i$. For example, where i=1, the set of shapes $C_0$ is searched for subshapes belonging to the subshape group $s_1$. Accordingly, as the counter i is incremented during the process flow, as described hereinbelow, the set of shapes to be searched ($C_{0...N-1}$) will be progressively searched for subshapes belonging to the other subshape groups until all the subshape groups are exhausted.

The set of shapes $C_{i-1}$ is searched for subshapes belonging to the group $s_i$ using the parametric transformations appropriate for that group. For example, for the default subshape group described hereinbefore with respect to Table 1 where i=1, the set of shapes $C_0$ is searched for subshapes of the group $s_1$ using translation, rotation, and isotropic scaling. Accordingly, where i=2, the set of shapes $C_1$ is searched for subshapes of the group $s_2$ using translation, rotation, and anisotropic scaling, and so on for the remaining subshape groups $s_3$ and $s_4$.

From block 46, the process continues to block 48, where it is determined whether a parametric transformation of a subshape belonging to the group $s_i$ is found in the set of shapes $C_{i-1}$. For example, where i=1, it is determined whether a parametric transformation of a subshape belonging to the group $s_1$ is found in the set of shapes $C_0$. If a subshape belonging to the group $s_i$ is not found in the set of shapes $C_{i-1}$, the process flow returns to block 32, where the operation of the parametric shape grammar interpreter 12 is terminated. The process flow is terminated at this point because a subshape belonging to the group $s_i$ is not found in the set of shape $C_{i-1}$, and if the subshape group $s_i$ is not null, then the left-hand shape of the selected rule cannot be found in the set of shapes $C_0$. Conversely, if at block 48 a parametric transformation of a subshape belonging to the group $s_i$ is found, then the process continues to block 50.

At block 50, a set of shapes $S_i$ is generated. The set of shapes $S_i$ includes the parametric transformations of the subshapes of the group $s_i$ found in the set of shapes $C_{i-1}$ using the transformations appropriate for that subshape group. For example, where i=1, a set of shapes $S_1$ is generated which includes the parametric transformations of the subshapes of the group $s_1$ found in the set of shapes $C_0$. For subshape groups that are null, the set $S_i$ is set to be a null, as described hereinbefore with respect to block 42.

Continuing to block 52, a set of shapes $C_i$ is generated which corresponds to the subtraction of the set of shapes $S_i$ from the set of shapes $C_{i-1}$. Thus, for example, where i=1, at block 52 the set of shapes $C_1$ is generated which corresponds to the subtraction of the set of shapes $S_1$ from the set of shapes $C_0$. For subshape groups that are null, the set $C_i$ is set to be the same as $C_{i-1}$, as described hereinbefore with respect to block 42.

From block 52, the process continues to block 54, where the set of shapes $S_i$ are added to the sum of sets $S_{i-1,\ldots,0}$. The set of shapes $S_i$ is added to the previous sum such that the connectivity of the decomposed left-hand shapes is maintained using, for example, the connectivity technique described herein. Thus, for example, where i=1, the set of shapes $S_1$ is added to the set of shapes $S_0$, which was set to null as described hereinbefore with respect to block 36. Accordingly, the sum of the sets $S_1$ and $S_0$ will be the same as $S_1$. The set $S_1$ will also be null if the group $s_1$ is null. Conversely, if $s_1$ is not null and if at block 48 parametric transformations of the subshapes belonging to the group $s_1$ are found in the set $C_0$, then the set $S_1$ will include those shapes corresponding to those parametric transformations. Accordingly, where i=2, the sum of sets $S_{2,1,0}$ will correspond to the sum of sets $S_2$ and $S_1$.

From block 54, the process flow continues to block 56, where it is determined whether i=N. This determination corresponds to a check of whether parametric transformations of the subshapes of each of the subshape groups $s_{i\ldots N}$ that are not null have been searched for.

If i does not equal N, then the process flow advances to block 58, where the connectivity of the subshapes of set $S_i$ relative to the set of shapes $C_i$, as well as the relative connectivity between the other parts of the decomposed left-hand shape, are determined. The relative connectivity of the parts of the left-hand shape may be determined by, for example, identifying with labels or weights the overlapping points of the subshapes of groups $s_1$, $s_2$, ..., $s_i$, and the

subshape of the next group that is not null. In addition, the points in the shapes of set $C_i$ corresponding in location to the transformed, identified points in the groups $s_1, s_2, ..., s_i$, may also be identified with, for example, labels or weights. From block 58, the process flow returns to block 44, where the counter (i) is incremented such that the shape recognition function may resume with the subshapes of the next subshape group.

It should be recognized that prior to advancement of the process flow to decision block 56, the set of shapes $C_i$ has been generated at either block 42 or 52, as described hereinbefore. At block 42, the set $C_i$ is set to be the set $C_{i-1}$ because the set $s_i$ is null. Accordingly, when the process flow returns to block 46 (assuming the group $s_{i+1}$ is not null), in essence the set of shapes $C_{i-1}$ will be searched for the subshapes of group $s_{i+1}$. Conversely, if at block 48, a parametric transformation of a subshape of the group $s_i$ was found in the set of shapes $C_{i-1}$, then the set of shapes $C_i$ is generated at block 52, as described hereinbefore, as the set of shapes $S_i$ subtracted from the set of shapes $C_{i-1}$. Accordingly, when the process flow continues to block 46, the set of shapes $S_i$ subtracted from the set of shapes $C_{i-1}$ (i.e., the set of shapes $C_i$) will be searched for subshapes of the group $s_{i+1}$ (again, assuming the group $s_{i+1}$ is not null).

If at block 56 it is determined that i=N, which corresponds to a determination that the presence of parametric transformations of subshapes belonging to each of the subshape groups $s_{i...N}$ which are not null have been searched for, then the process flow proceeds to block 59, where the sum of sets $S_{i...N}$, as determined at block 54, corresponds to the parametric transformations of the left-hand shape of the selected rule found in the set of shapes $C_0$.

According to other embodiments of the present invention, the interpreter 12 may recognize parametric transformations of the left-hand shape of a selected rule according to process flows different than that illustrated in Figure 4. For example, according to another



16

embodiment, rather than adding the set of shapes $S_i$ to the sum of $S_{i-1...0}$ at block 54 prior to the determination of whether i=N at block 56, the sets $S_{i...N}$ may be summed together in one step after the determination of whether i=N to recognize the parametric transformations of the left-hand shape of the rule in the set of shapes $C_0$.

Once the parametric transformations of the left-hand shape of a selected rule is recognized in the set of shapes $C_0$ by the parametric shape grammar interpreter 12, as described hereinbefore with reference to Figure 4, it may be determined whether the rule is to be applied to the set of shapes $C_0$. This determination may be made, for example, by an operator of the system 10 or the intelligent rule selection module 20. If a particular application of the rule is selected, the rule application module 18 may then apply the rule by subtracting the transformation of the left-hand shape of the rule from the initial shape and adding a transformation of the right-hand shape. After the rule is applied, the process flow illustrated in Figure 4 may be repeated with the selection of a different rule from the set of predefined rules to be applied to the resulting shape (or shapes) from the application of the prior rule. If it is determined that the rule is not to be applied, the process flow illustrated in Figure 4 may also be repeated with the selection of a new rule from the set of predefined rules to be applied to the original shape or shapes ($C_0$). According to another embodiment, the rule application module 18 may apply the rule for all transformations of the left-hand shape found in the set of shapes $C_0$, and the process may be repeated for all of the resulting shapes, thus producing all possible permutations resulting from application of the predefined set of rules in the initial design shape(s).

The I/O interface module 22 may be used to input data, such as the shape grammar rules, and to output data, such as the set of rules, the transformations of the left-hand shape of a particular rule found in a shape, and the shapes resulting from the application from a particular

rule. The I/O interface module 22 may input and output the data, for example, in text and/or graphical form. The I/O interface module 22 may display data via a display device (not shown) in communication with the I/O interface module 22.

Thus, the parametric shape grammar interpreter 12 of the present invention permits parametric shape recognition of the left-hand shape of a shape grammar rule in an initial design shape(s). Unlike previous interpreters that are limited to Euclidean transformations (translation, rotation, and scaling) that can only be applied to whole shapes, the parametric shape grammar interpreter 12 can search for general parametric features of a subshape generated through decomposition of a shape, thus allowing for separate treatment of each subshape.

Figures 5-11 provide a shape decomposition example using the example default hierarchy of subshape groups defined hereinbefore with respect to Table 1. Consider the shape to be decomposed (such as the shape $a$ in the rule $a \rightarrow b$) to be that illustrated in Figure 5. To recognize the transformations of the subshapes of the groups $s_{1\text{-}4}$, as defined hereinbefore, the lines of symmetry in the shape of Figure 5 may first be determined. These lines of symmetry are illustrated in Figure 6 as dashed lines. As illustrated in Figure 6, each line of the square 60 is symmetric with the two lines of the square 60 that it intersects. In addition, each of the lines of the triangle 62 is symmetric with more than one line. Accordingly, these subshapes satisfy the requirements of the subshape group $s_1$, and can be subtracted from the example shape, resulting in the shape shown in Figure 7, for which the subshapes of group $s_2$ may be searched.

The resulting shape, shown in Figure 7, contains two lines that are symmetric to only one other line. Additionally, there are two perpendicular intersections, comprised of three line segments, that satisfy the requirements of $s_2$, as illustrated in Figure 8. Accordingly, this shape



18

may be subtracted from the shape shown in Figure 7, resulting in the shape shown in Figure 9, which may be searched for subshapes of the group $s_3$.

The $s_3$ subshape illustrated in Figure 10 is present in the shape of Figure 9. As illustrated, the $s_3$ subshape is simply the intersecting line segments. Accordingly, this subshape may be subtracted from the shape of Figure 9, resulting in the shape shown in Figure 11, which corresponds to the subshapes comprising the $s_4$ group.

Figures 12-19 provide an example of parametric shape recognition, using the example default hierarchy defined hereinbefore with respect to Table 1, to recognize the presence of parametric transformations of the left-hand shape ($a$) of the rule ($a \rightarrow b$) in a design shape ($C_0$). Consider the rule to be the rule $a \rightarrow b$ illustrated in Figure 12, and consider the initial design shape ($C_0$) to which the rule is to be applied to be the shape illustrated in Figure 13. As described hereinbefore, in order to apply the rule $a \rightarrow b$ to the design shape $C_0$, the left hand shape ($a$) of the rule must be found to be a parametric subshape under various transformations ($\tau$) of the shape $C_0$. Using the default hierarchy defined hereinbefore with respect to Table 1, the shape $a$ may be decomposed into the four subshapes where $a = s_1 + s_2 + s_3 + s_4$.

For the shape $a$ shown in Figure 12, using the default hierarchy defined hereinbefore with respect to Table 1, the subshapes comprising groups $s_1$ and $s_2$ are shown in Figure 14, and the groups $s_3$, $s_4$ are null. The shape recognition process, as described hereinbefore, may begin with the most constrained subshape group that is not null and skipped any less constrained groups that are null. Such an embodiment produces a more efficient shape recognition process because the more highly constrained shapes have fewer possible transformations. Thus, for the rule shown in Figure 12, the $s_1$ subshape is searched first, and then the $s_2$ subshape is searched.



19

Permissible transformations of the $s_1$ subshape may be found multiple times in the shape *a*, resulting in four instances of $s_1$ subshapes in this example. These transformations, as described hereinbefore, are defined as the set $S_1$, and are shown in Figure 15. The four shapes of $S_1$ are equal but are found differently within the initial design shape by the rotation of $s_1$ subshape four different ways ($0^0$, $90^0$, $180^0$, and $270^0$). The dots in Figure 15 are to show the various transformations of the $s_1$ subshape found in the shape *a*. Having found the set of shapes $S_1$, the set of shapes $C_1$ is generated, which is the result of the set of shapes $S_1$ subtracted from $C_0$. The set of shapes $C_1$ is shown in Figure 16.

By definition of the subshape groups $s_1$, $s_2$, $s_3$, and $s_4$, it can been seen that no two groups will share any common line segments. They will, however, share common line segment end points. Accordingly, the relative connectivity of the shapes of groups $s_1$ and $s_2$, as well as the relative connectivity of the transformed instance of $s_1$ and the set of $C_1$ shapes may be identified, as illustrated in Figure 17.

Next, as described hereinbefore, the set of shapes $C_1$ is searched for the next most constrained subshape group, which for this example, is the $s_2$ group. As can be appreciated, two permissible transformations of the $s_2$ subshape may be found in each of the shapes of $C_1$. The set of the subshapes thus define the set $S_2$. Next, as described hereinbefore, the set of shapes $S_2$ is subtracted from the set of shapes $C_1$ to define the set of shapes $C_2$. Next, the intersection points between the marked shapes $S_2$ and the corresponding shapes $C_2$ are identified.

The sets $S_1$ and $S_2$ are then added such that their connectivity is maintained to produce the subshapes illustrated in Figure 18. Because the groups $s_3$ and $s_4$ are null, as described hereinbefore, the shapes illustrated in Figure 18 represent the parametric transformations of the



21

left-hand shape $a$ of the rule $a \rightarrow b$ (illustrated in Figure 12) found in the initial design shape $C_0$ (illustrated in Figure 13). The two possible applications of the rule may then be applied to the shape $C_0$ to produce the shapes illustrated in Figure 19.

Figures 20-23 provide an example of parametric rule application. Consider the rule to be applied as the rule $a \rightarrow b$ illustrated in Figure 20, and the initial design shape $C_0$, to which the rule is to be applied, as the shape illustrated in Figure 21. Using the default hierarchical subshape groups described hereinbefore with respect to Table 1, it can be recognized that the left-hand shape ($a$) of the rule has constraints that limit the parametric shape search to perpendicular intersections. This corresponds to group $s_2$. Twelve permissible transformations of the $s_2$ shape may be found in the shape $C_0$, three of which are shown in bold in Figure 22. Because the subshape groups $s_1$, $s_3$, and $s_4$ are null for this example, the sum of sets $S_{1\text{-}4}$ includes only the twelve transformations of the $s_2$ subshape found in the shape $C_0$. Accordingly, the shape $a$ may be recognized twelve times in the shape $C_0$, with application of the rule for each of the transformations resulting in the shapes illustrated in Figure 23.

Figures 24-27 provide another example of a parametric shape grammar application using the default hierarchy of subshape groups described hereinbefore with respect to Table 1. For the example, the set of rules illustrated in Figure 24 comprise the predefined shape grammar rules, and the initial design shape is the shape illustrated in Figure 25. Upon examining each of the rules, it can be recognized that the left-hand shapes of each rule fall into the $s_3$ group because of the lack of symmetry and perpendicular intersections. Therefore, in general, each of the rules may be applied if a shape corresponding to a permissible parametric transformation of the left-hand shape of any of the rules is recognized in the initial design shape. For example, rule 1 is



21

applicable if any triangle can be recognized, and rule 4 may be applied if any five-sided polygon can be recognized. The progression of shapes illustrated in Figure 26 depict the application of a series of these rules using the parametric shape grammar interpreter 12 for shape recognition. For the shapes illustrated in Figure 26, the subshape to which the indicated rule is to be applied is highlighted in bold. The progression of rule application may continue, such as by randomly choosing the applicable rules as well as the parameters, producing final design shapes such as those illustrated in Figure 27.

Those of ordinary skill in the art will recognize that many modifications and variations of the present invention may be implemented. The foregoing description and the following claims are intended to cover all such modifications and variations.



22